

# Exam 2 Review

CS461/ECE422 Fall 2007

# Exam guidelines

- A single page of supplementary notes is allowed
- Closed book
- A calculator is allowed.
- Students should show work on the exam. They can use supplementary sheets of paper if they run out of room.
- Students can use scratch paper if desired.

# Exam logistics

- Exam will be given in the evening (7-8:15pm) in 112 and 114 of the Transportation Building
  - <http://webtools.uiuc.edu/ricker/CampusMap?target=s>
- Students will be split by last name
  - Go to the same room you where in last time
  - A-K in one room and L-Z in the other

# Topics

- Memory protection
- Discretionary access controls
- Authentication
- Trusted OS Policies and Models
- Trusted OS Features and design
- Evaluation
- Malware and commonly exploited program errors
- Secure development techniques

# Memory protection

- Techniques to separate User from System programs and users from each other
  - Vary in flexibility and efficiency
  - Fence registers, base-bounds registers, page tables, segment tables, paged-segment tables
- Hardware techniques
  - Rings – separate high and low privilege processes
  - Read, Write, Execute bits

# Access Control

- Access Control Matrix
  - General model that captures the expression of subjects, objects, and rights
- Access Control Lists
  - Column oriented approach
  - Access information stored with objects
- Capabilities
  - Row oriented approach
  - Access information stored with subjects
- Role-based

# Transitions in ACM's

	Obj1	Obj2	Obj3	Obj4	Alice	Bob	Carol	Dave
Alice	R	RWX		RW		RW		
Bob		RW			RW		R	
Carol	RW		RWA			W		
Dave				RWA				

# Capabilities

- Rights allowed to particular objects stored with object
  - In Hydra OS, all objects references were handles that included the capabilities, stored in kernel space
  - In other systems, capabilities were stored in typed memory
- Could pass capabilities to other processes
  - Temporary proxy or delegation
  - Problems bounding delegation
- Online book on early capability systems
  - <http://www.cs.washington.edu/homes/levy/capabook/>

# Authentication

- Establish ID
  - What you know
  - What you have
  - What you are
- Spent a lot of time on passwords
  - On line vs off line attacks
  - Salt
  - Anderson's formula
- Challenge Response
- Biometrics

# Trusted Models and Policies

- Mandatory Access Control (MAC)
- Bell LaPadula
  - Confidentiality Policy
  - Lattice of security labels, e.g., Security:{Proj1, Proj2}
  - Read down, write up
- Strict Biba
  - Integrity Policy
  - Dual of Bell LaPadula

# More Trusted Policies and Models

- Clark-Wilson Integrity Model
  - Enforcement and certification rules to guide creation of high integrity system
  - Track which programs are certified to access which data
  - Track which users are allowed to invoke programs on which data sets
- Chinese Wall
  - Control Conflict of Interest Concerns
  - Dynamic
  - Write restrictions too severe to be practical

# Trusted System Design

- Design Principles of Saltzer and Schroeder
- Key Security Features
  - MAC
  - Identification and Authentication
  - Object Reuse Protection
  - Complete Mediation
  - Trusted Path
  - Audit
- Trusted computing base and kernelized design

# System Evaluation

- Separation of Security functions and Security Assurance
- TCSEC or orange book
  - Fixed set of classes with specific functionality and assurance requirements
- Common Criteria
  - Fixed sets of assurance requirements EALs
  - Dynamic and evolving functionality targets
    - Security targets and protection profiles

# Example Trusted OS

- Attempt to give example of issues encountered in real implementations
  - Multi-level directories (for BLP)
  - Least Privilege Granularity
    - For all systems
    - Including “general OS”
      - Forgot to mention Linux “capabilities” and Windows privileges
  - Constraining privilege
    - Means to express privilege amplification and reduction on process invocation

# Example Trusted OS

- Not all MAC models are BLP
  - PitBull LX
    - Category only label
    - Read and write access allowed if subject has superset of object categories
  - SE Linux
    - Type Enforcement
    - Subjects and objects are labeled with types
    - Arbitrary allow statements define access based on types
  - AppArmor
    - Types associated by file path
    - Define profile for program

# Malicious Code

- Zero Day exploits
- Virus
- Rootkits
- Trojans
  - Thompson's deeply hidden example
- Worms
- Covert Channels

# Commonly Exploited Errors

- Buffer Overflows
- TOCTTU
- Poor input checking
  - Cross Site Scripting attacks
- Some means to detect errors in programs
  - Software fault injection
  - Fuzzing
  - Penetration testing

# Secure Development

- Good software engineering
  - Modular Design
  - Identify Security Architecture
  - Refine requirements
- Secure Development
  - Peer review
  - Threat modeling
  - Testing – in general and security in particular
  - Configuration management

Good luck!

